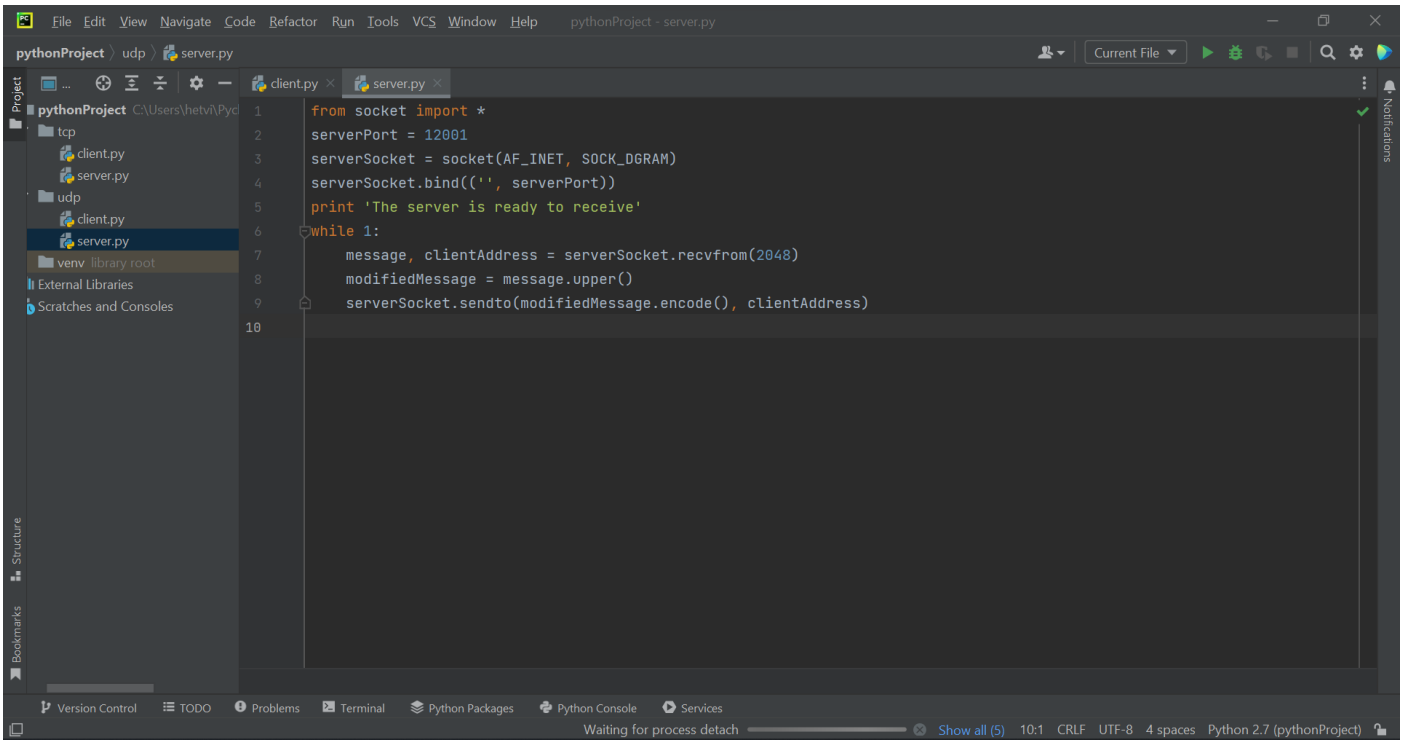
NAME: Hetvi Patel **CSC 138**
DATE: 10/05/2022

Socket 1

UDP Screenshots :

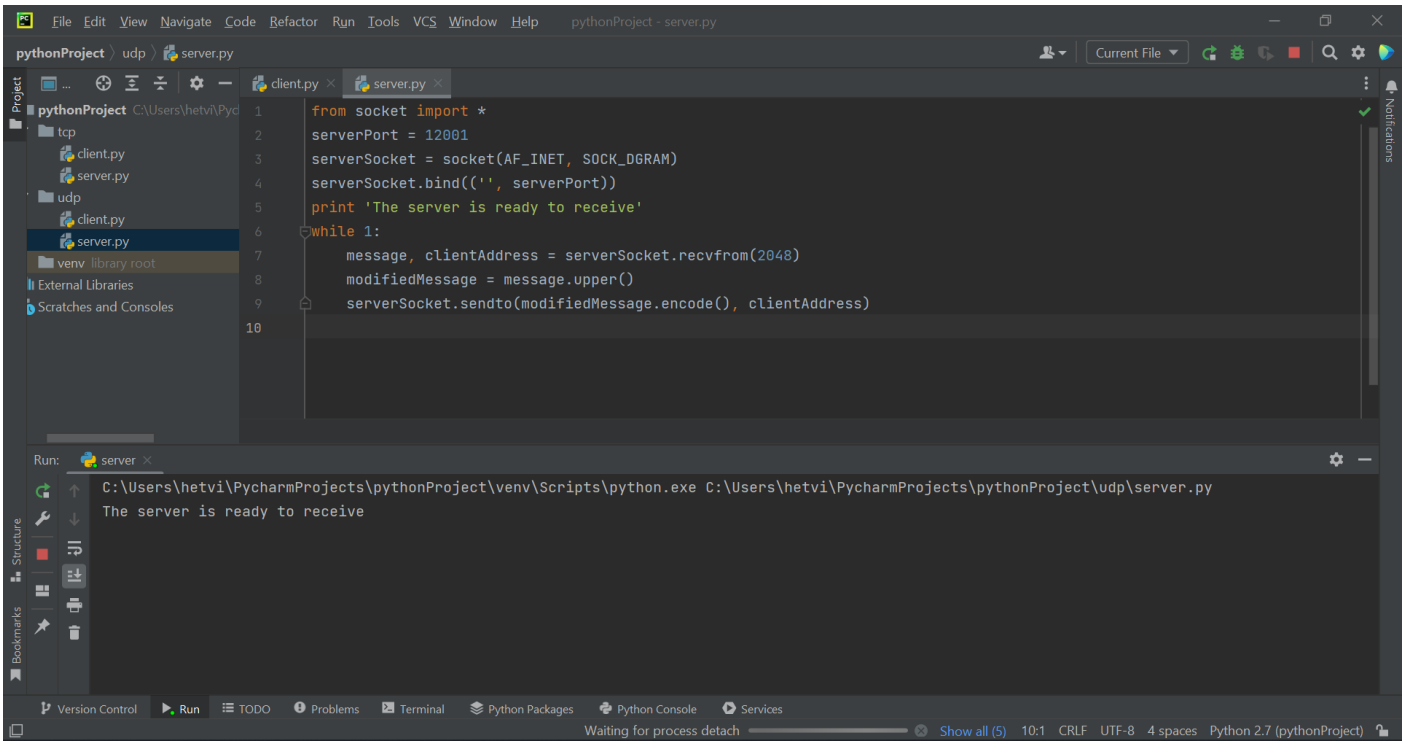
Source code for server:

```
from socket import *
serverPort = 30002
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', serverPort))
print ('The server is ready to receive')
while 1:
    message, clientAddress = serverSocket.recvfrom(2048)
    modifiedMessage = message.upper()
    serverSocket.sendto(modifiedMessage.encode(), clientAddress)
```



This is the source code for udp server where it prints out 'The server is ready to receive'. It takes the modifiedMessage and changes it to uppercase letters. The server never requires a close() function.

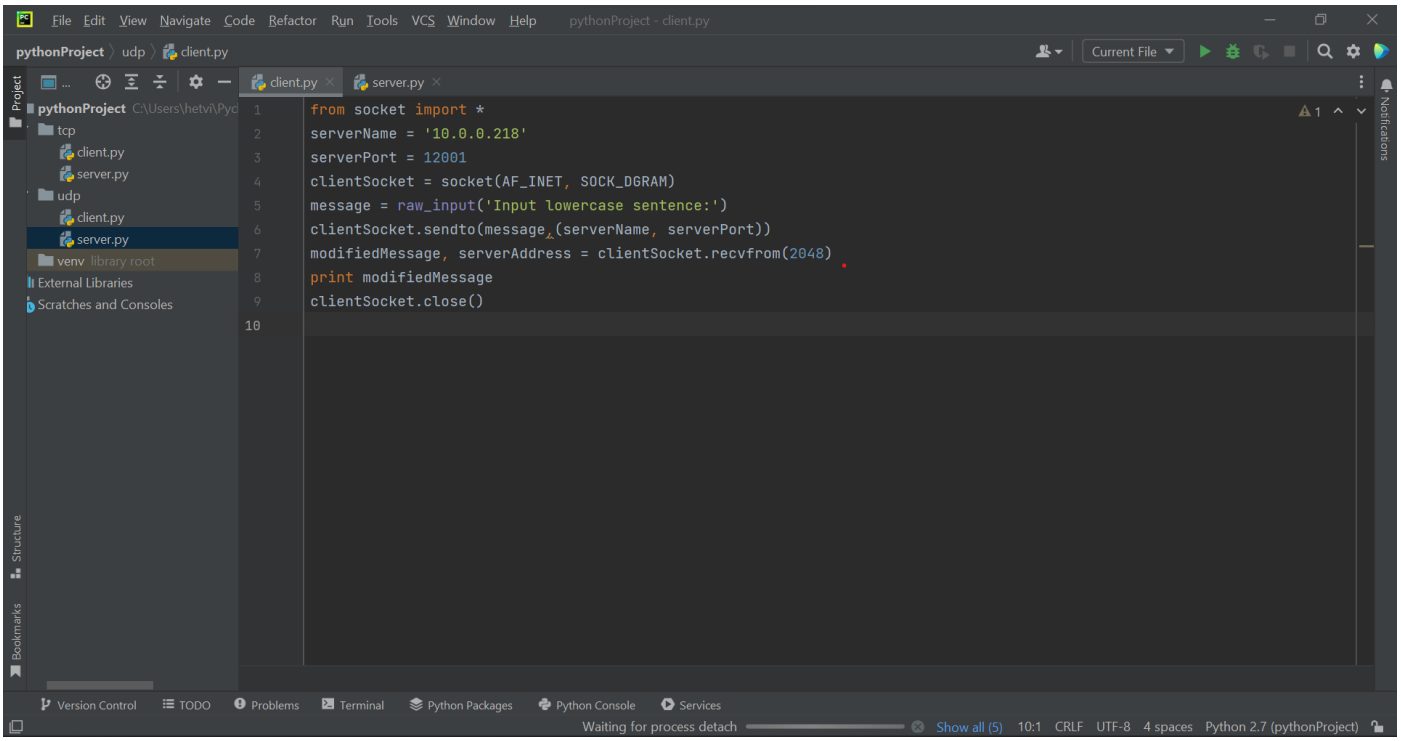
Output for server:



This is the output for udp server where it shows that the server is ready to receive the message from client.

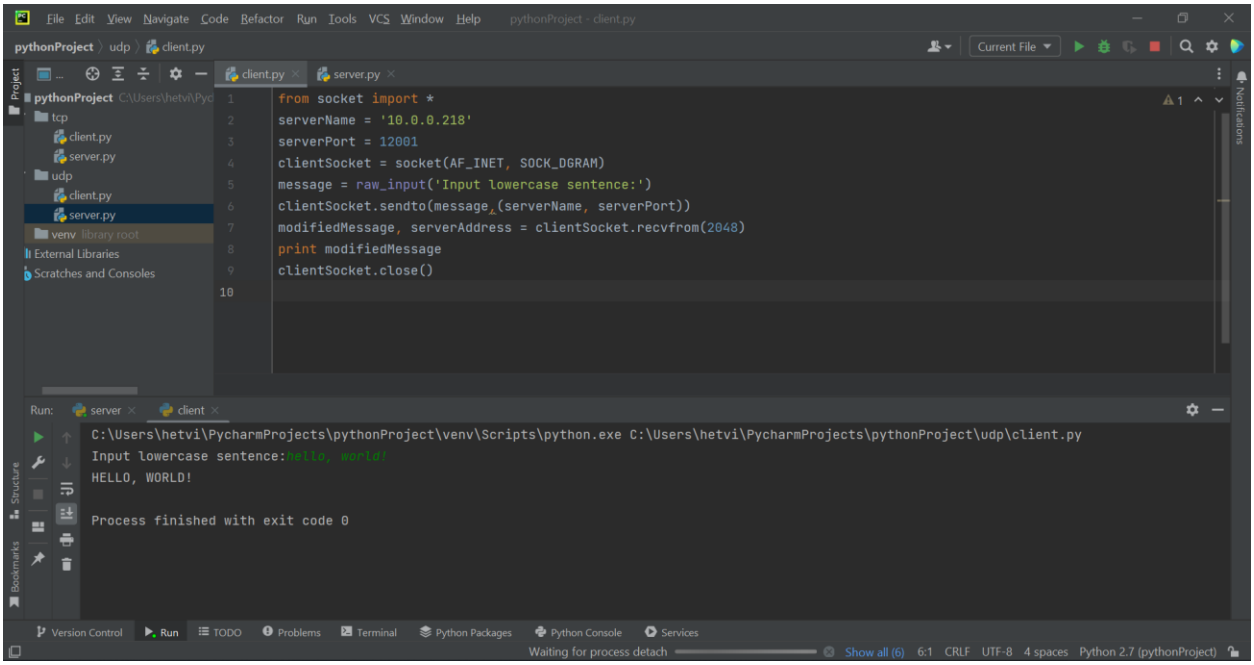
Source code for client:

```
from socket import *
serverName = '10.0.0.218'
serverPort = 30002
clientSocket = socket(AF_INET, SOCK_DGRAM)
message = raw_input('Input lowercase sentence:')
clientSocket.sendto(message,(serverName, serverPort))
modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
print modifiedMessage
clientSocket.close()
```



This is the source code for udp client where it asks to input a lowercase sentence which will then output that sentence as an uppercase sentence using udp server. The udp client file always uses close() function.

Output for client:



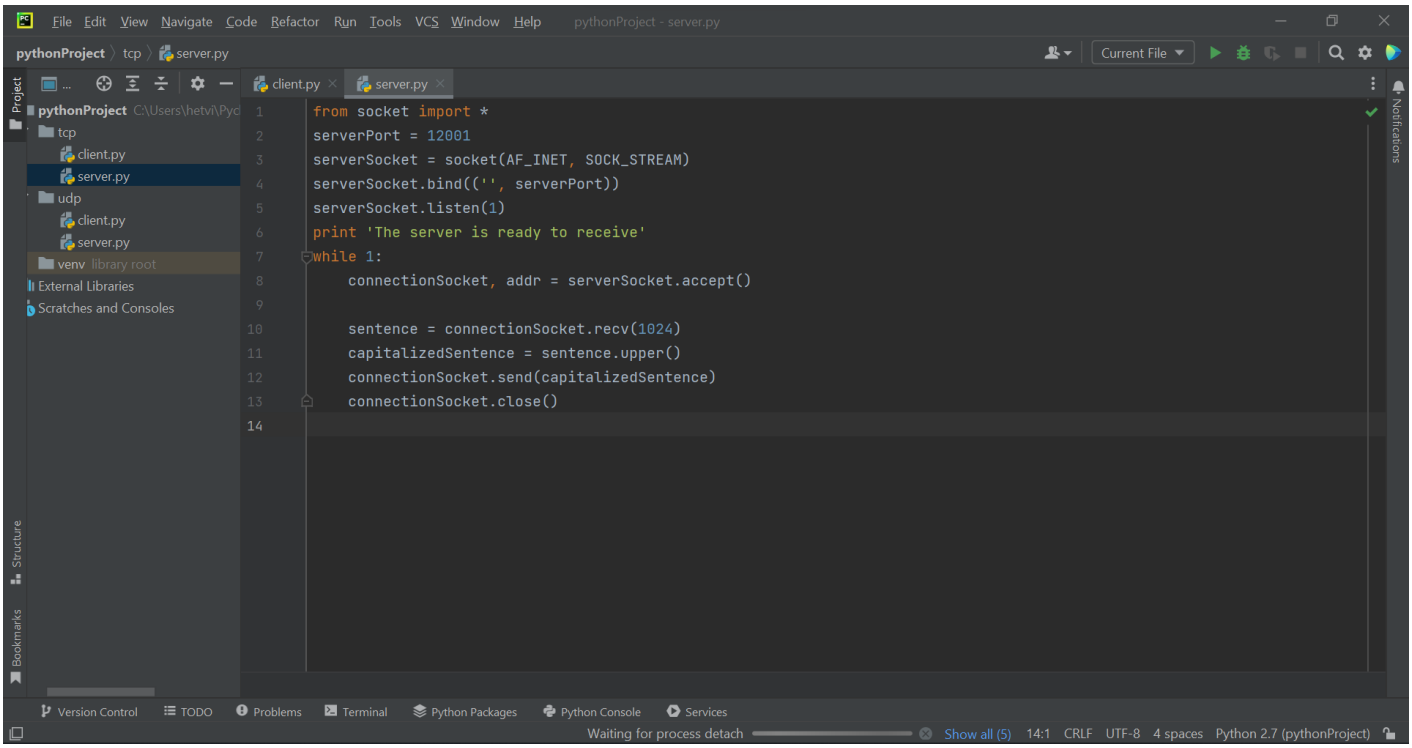
This shows us the output for the lowercase input sentence converted to the uppercase sentence.

TCP Screenshots :

Source code for server:

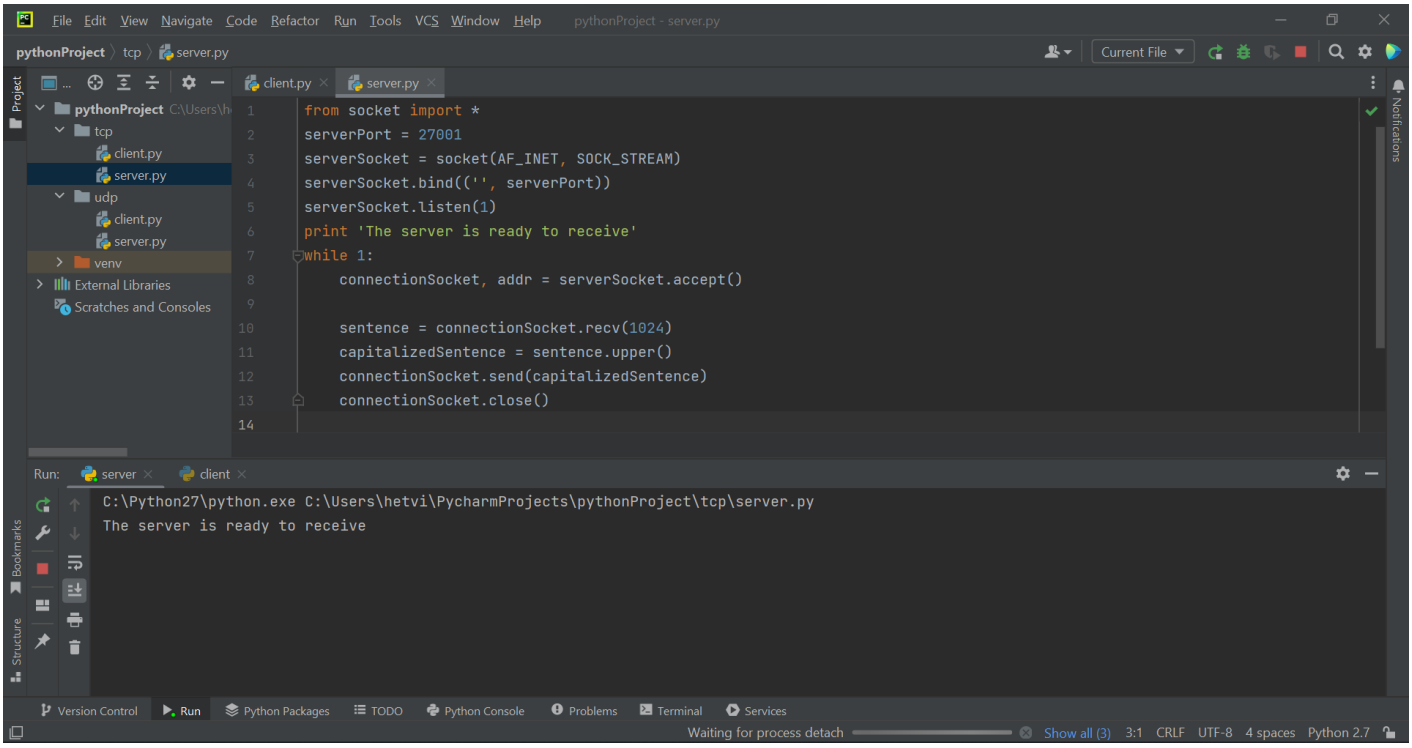
```
from socket import *
serverPort = 27001
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('', serverPort))
serverSocket.listen(1)
print 'The server is ready to receive'
while 1:
    connectionSocket, addr = serverSocket.accept()

    sentence = connectionSocket.recv(1024)
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence)
    connectionSocket.close()
```



This is the source code for tcp server where it prints out 'The server is ready to receive'. The server never requires a close() function. Here only connectionSocket is using close() function because it needs to close for server to output.

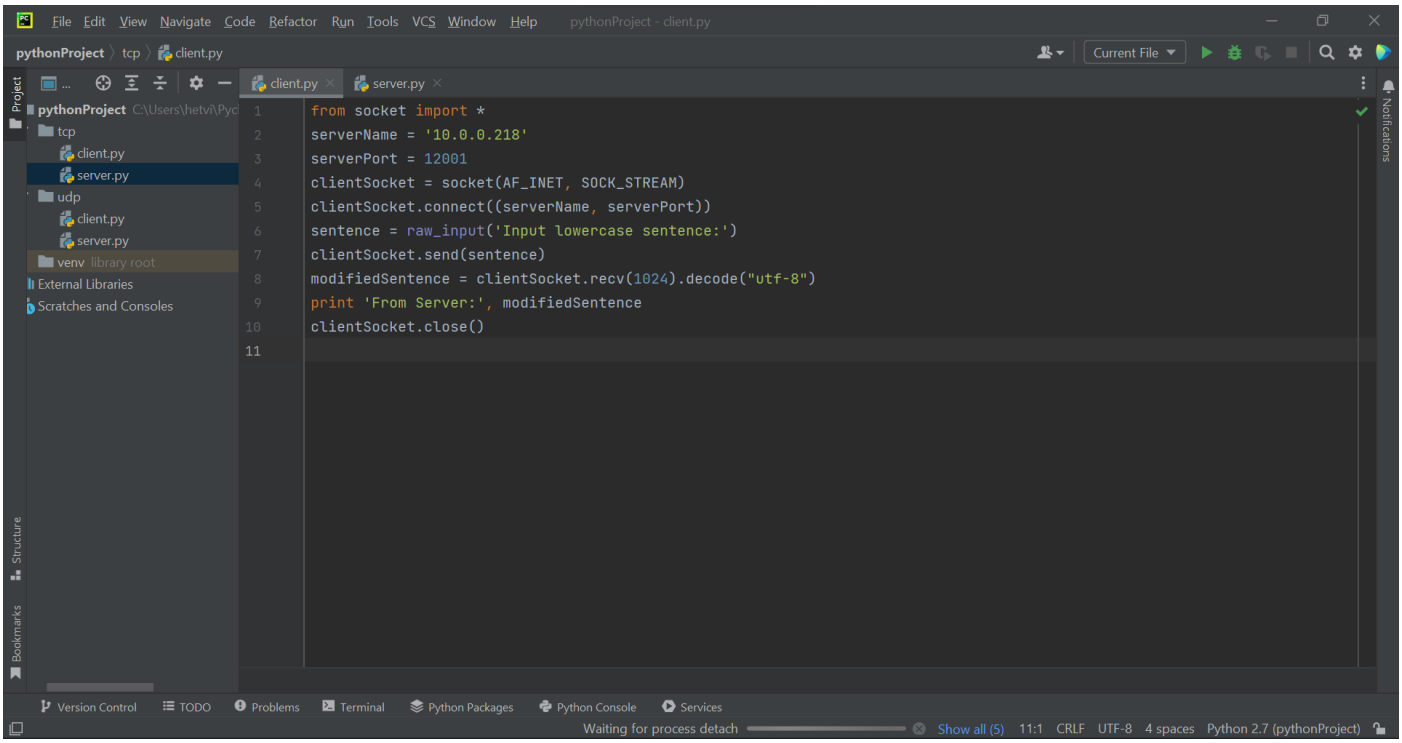
Output for server:



This is the output for tcp server where it shows that the server is ready to receive the message from client.

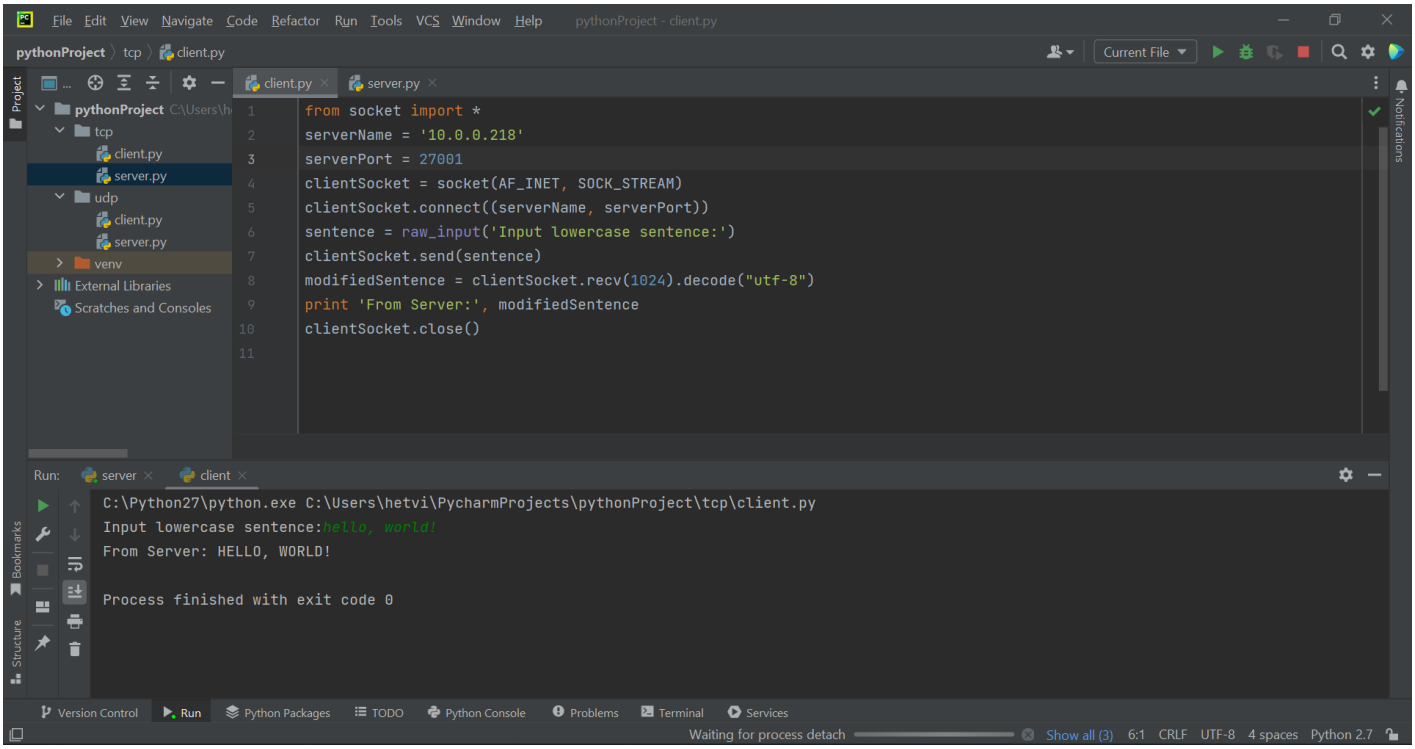
Source code for client:

```
from socket import *
serverName = '10.0.0.218'
serverPort = 27001
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = raw_input('Input lowercase sentence:')
clientSocket.send(sentence)
modifiedSentence = clientSocket.recv(1024).decode("utf-8")
print 'From Server:', modifiedSentence
clientSocket.close()
```



This is the source code for tcp client where it asks to input a lowercase sentence which will then output that sentence as an uppercase sentence using tcp server. The tcp client file always uses close() function.

Output for client:



This shows us the output for the lowercase input sentence converted to the uppercase sentence.